

AFOSR-TR. 89-0781

Final Report
Grant No. AFOSR-87-0065
DOD-URIP



**Testbeds for Logic Programming
and Very Large Databases**

Principal Investigator:

Prof. Kenneth A. Bowen

**Logic Programming Research Group
School of Computer & Information Science
313 Link Hall
Syracuse University
Syracuse, NY 13210**

Accession For	
NTIS GRA&I	X
DTIC TAB	
Unannounced	
Justification	
By	
Distribution	
Availability Codes	
Dist	
A-1	

Goal of the Instrumentation Request:

Support of research in advanced logic programming and its application to artificial intelligence, primarily in the following areas:

- Extensions and enhancements to the Prolog approach to logic programming, including metalevel programming capabilities, alternative sequential control, and co-routining and concurrent execution of logic programs;
- Interfaces to very large databases maintained by software or specialized hardware;
- Applications to the maintenance of very large knowledge bases and to the construction of expert systems.

Original Equipment Request:

- Two (2) Sun 3/160M-8 workstations with 380MB Disks and 1600 bpi 1/2" tape drives;
- One (1) Motorola C3I workstation testbed, including 68020 (12.5 MHz) processor, 160MB disk, 4MB RAM;
- Three (3) Xenologic Prolog Accelerator boards for installation in each of the above workstations;

Reasons for revision of equipment request:

- Inability to utilize Xenologic Prolog Accelerator boards for planned

research;

- Poor cost-effectiveness of Motorola C3I workstation testbed.

Revised Equipment Configuration (acquired):

- One (1) Sun 3/280-16 rack-mounted server with two (2) 575MB disks and one (1) 6250 bpi 1/2" tape drive;
- One (1) Sun 3/260C-8 processor with deskside 380MB disk;
- Four (4) Sun 3/75M-8 diskless processors;
- One (1) Sun 3/52M-4 processor with 71MB disk;
- One (1) Sun 3/50M-4 diskless processor;
- Printers & bit-pad;
- Kinetics Fastpath Gateway and 2 Macintosh computers;
- System, database, and utility software.

When the original URIP request was submitted, discussions with Xenologic had led us to believe two key items:

- The design of the installed microcode on the boards would include certain extensions to the abstract Prolog machine instruction set which would be directly oriented towards experimentation with the concepts underlying the metaProlog project;
- The effort necessary to install further specialized instructions would be moderate in scope.

Neither belief turned out to be correct. Not only did Xenologic decide to forego the anticipated extended instructions, but the microcode space on the board was totally consumed by their design. Thus, adaptation of the boards to directly support metaProlog implementations would have required a complete redesign and reimplementing of the microcode, which was unthinkable, given our resources. Moreover, we observed that Xenologic had considerable difficulty in implementing their own designs, falling far behind their original projected schedules. It was obvious that since we had no staff with extensive microcode design and implementation skills, entering into any such venture would have been an unmitigated disaster.

Consequently, we returned to our original conception which had evolved before consideration of the Xenologic boards deflected our planning. This was to utilize the programming language C to implement an appropriate underlying abstract machine for development of the metaProlog system. This had at least three points in its favor:

- C is a well-understood and relatively easy-to-use system level implementation language; Moreover, our staff had considerable experience in its use; Consequently, we would be able to implement and experiment with our designs relatively easily;
- C compilers produce quite efficient code; Consequently, if we ran on relatively powerful workstations, we could expect to be able to carry out reasonably complex experiments with the resulting metaProlog system;
- By utilizing C on a standard workstation, we would be able to make the resulting systems widely available to the research community for experimental purposes.

These arguments, the wide-spread use of UNIX, and the well-known arguments in favor of powerful personal workstations coupled via local area nets to a central server firmly convinced us. The remaining question was the specific choice of manufacturer. We felt it was important to choose equipment which maximized our compatibility with other university and industrial research groups, relying on three additional arguments:

- This would make distribution and wide use of the resulting systems much easier;
- The equipment and software would be maximally reliable, having experience wide and intense use;
- We would be able to acquire and install significant amounts of related research software developed by other similar groups.

This left us with a choice between a Sun Microsystems workstations and Digital Equipment Corporation microVAXs. Price together with the availability of UNIX on the Suns led us to decide in favor Sun, acquiring the equipment listed above.

The only negative aspect of this choice lay in the difficulty in actually acquiring the equipment, which we did not anticipate would take so long as it turned out. Because of the regulations concerning grants, we were unable to place a firm order for the equipment with Sun until 1 October 1986. Because of the high demand for Sun equipment, the server was not

received and installed until February of 1987, a delay of nearly five months. We also had difficulty in obtaining space from the university to house the equipment and provide workspaces for the staff. However, this was resolved shortly before the server was shipped.

Effect on the Research Project

Briefly put, the effect of this equipment on our research project was profound. Once the equipment was completely installed, our work proceeded at a considerably increased rate. Not only were we able to pursue the original scope of the metaProlog project, but we were able to with significant issues concerning the relations to relational databases and distributed systems. Specifically, we have accomplished the following:

- We completed the design and implementation of two versions of an initial metaProlog compiler system providing a significant subset of the theoretical capabilities of metaProlog;
- We have been able to significantly extend one of the initial versions of metaProlog to capture a much larger subset of the theoretical capabilities;
- We have been able to carry out a number of moderate scale knowledge base maintenance experiments utilizing the metaProlog implementations;
- We have been able to implement a large sophisticated expert system which relies heavily on the metaProlog facilities;
- We were able to develop methods for directly coupling Prolog and metaProlog programs to external relational databases in a very flexible and powerful manner;
- We were able to develop methods for permitting independent Prolog and metaProlog programs running on separate machines to communicate with each other via the local area network in an efficient high-level manner;
- We were able to develop powerful methods of providing Prolog and metaProlog programs with control of the SunView windowing interface, thus providing powerful user interfaces for applications programs;

We have published, and are continuing to publish, numerous papers on the metaProlog work (which have been sent to AFOSR), and have received a substantial number of requests for copies of the systems. We are currently

arranging to being distribution of the systems to the research community.

The work accomplished under the metaProlog project far exceeded that described in the original URIP proposal. While the work of the related project directed by Prof. Bruce Berra has moved forward, the points discussed in the URIP proposal have not been reached. Specifically, that project has not yet reached the stage of development of specialized hardware which would interface with the metaProlog system via a VME bus. (However, an Australian group has developed an inexpensive board built from off-the-shelf components which addresses many of the same questions. We exploring acquiring this board.) However, as indicated above, the metaProlog group alone has addressed the major issue being pursued by Berra's group, namely the provision for coupling logic-based systems with large knowledge bases. As indicated, we have developed software techniques for directly coupling logic-based systems to external relational databases such as ORACLE (which we used in the work). These techniques allow us to couple expert systems programs and knowledge base maintenance programs simultaneously with multiple relational databases maintained by a heterogeneous collection of database management systems. From the point of view of the logic program, the data in the relational databases appears to be a large collection of Prolog facts. The program can match the facts, add to (assert) the facts, and remove (retract) facts exactly as it does for ordinary facts. That is, the interface makes the relational tables appear to the logic program as a virtual set of facts — the logic program effectively cannot tell that the facts are external to the system. Thus, large sets of facts can be incorporated in the logic programs. By judicious design, the ability to simultaneously access multiple external databases can be used with an intelligent "divide-and-conquer" approach to provide logic programs with access to very large knowledge bases. Moreover, utilizing the remote communications facilities we have developed, the databases accessed by the logic program can reside on remote machines.

Also, it is the nature of the interfaces that the relational database is not affected at all: The interfaces appear as just another external application running over the database. Consequently, logic-based expert systems can access data residing in existing large databases on a current up-to-date basis, which is essential for many expert applications. Since DoD possesses many extremely large databases which are potentially involved in expert systems applications, this is a significant point. Moreover, these

databases already have many applications running over them which must continue to function after a new expert application is installed. Thus it is not thinkable to convert these databases to specialized formats designed for expert systems applications. Our approach appears to have cut through a tangle of conflicting requirements to provide an extremely attractive solution to knotty problems.

It is our opinion that the research work has been highly fruitful, and that the URIP grant support was essential in accomplishing this work.